Plaintiffs' Exhibit 122

Executive Summary

Proprietary + Confidential

- **New:** We launched an internal Header Bidding dashboard available at go/hbmonitor accessible to all Sell-Side teams.
- **HB adoption** continues to **grow** globally, now reaching **50%** and is projected to reach **66%** by EOY 2018. HB **impression growth** is mainly driven by **NA** (+15B YoY) & **EMEA** (+10B YoY).
- **Non-HB 3rd party indirect** keeps **losing ground** to AdX, Exchange Bidding & Header Bidding. **HB** impressions grow **faster** than AdX impressions in percentage, but **AdX** still **grows more** than **HB** in terms of **absolute volume**.
- Each HB competitor is searching for **differentiation** (pricing/fees, supply path optimization & server side wrappers) but globally **Amazon** has experienced the **fastest HB growth** in the past 12 months. Also the HB market became more **fragmented** in H2 2017 with **Rubicon** & **OpenX** losing share.
- Some **HB implementation** in DFP can be **sub-optimal**, mainly due to low AdX eligibility, inadequate HB LIs Reservation Type, or too wide rate granularity. In order to **optimize** these implementation and ensure **fair competition** with **AdX,** we sent 50+ proactive (representing **$49m+** uplift opportunity) and 120+ ad hoc HB reports to AMs in Q4 2017 & Q1 2018. This proactive outreach increased AdX eligibility by **4%** points and AdX win rate by **3%** points for the targeted accounts.
- We also launched a new service providing **proactive HB detection** surfacing publishers who **recently enabled HB** for the first time to their account managers.
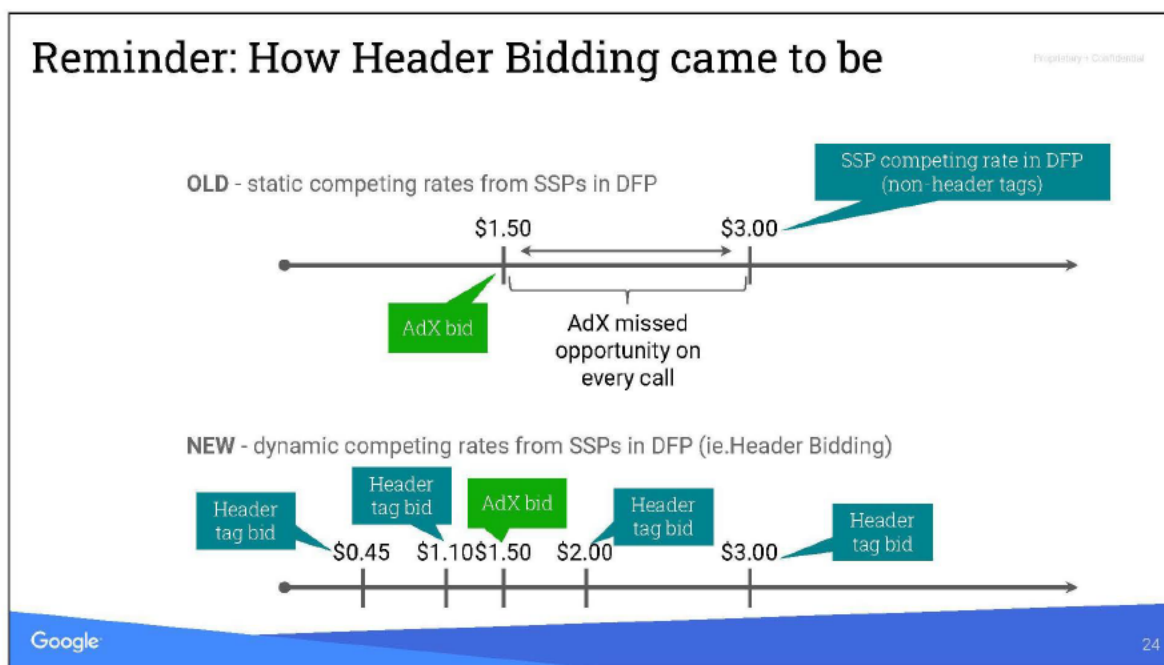
Google

Source: DRX data, LPS + OPG T1, DAOF only

2

Header Bidding Refresher

Google

Let's quickly review how header bidding came to be.

Back when AdX launched, we imagined publishers would select one exchange partner to manage all programmatic demand. Managing multiple ad networks with different static price points had proven to be such a headache for pubs it didn't seem logical that they would want to go down the same road with exchanges, especially since all exchanges had access to essentially the same demand. That was except for AdX, which had exclusive access to GDN demand. Exclusive access to the largest ad network in the world and the fact we had our proprietary contextual targeting algorithm, we assumed that AdX would be the preferred exchange for pubs. However, it quickly became apparent pubs were willing to work with multiple exchanges. Our clean second price auction was perceived by some as leaving money on the table, due to the gap between a marketer's willingness-to-pay and the auction closing price. But in a second price auction, buyers will generally bid higher than what they are willing to pay, because they have the expectation that the clearing price will be lower. Pubs and other Exchanges saw this as an opportunity to make more money. If a competitive exchange could close an auction at a higher price through a modified second price auction they could increase yield for a pub. In short, pubs came to realize pitting multiple exchanges against one another fostered price competition, which was good for business.

Of course, throughout this period AdX was not our only product in the market. We were committed to our 'own the tag' DFP strategy and our full stack capabilities still put us significantly ahead of our ad tech peers. Dynamic Allocation was a key strategic pillar for us. The fact that AdX demand could bid and win in real-time against our tag-based indirect competition and ultimately against directly sold ads was key to establishing ourselves as a preferred holistic yield management partner to a great many pubs across the globe. We proved that being able to submit prices on a per impression basis was good for pubs. This became even more apparent as user targeting such as our GDN and DBM interest category and remarketing capabilities grew to billion dollar plus businesses.

Other exchanges booked within DFP did not feel it was fair that AdX was given special per impression treatment. AdX network buying partners that enjoyed the benefits of Dynamic Allocation in DFP wanted to make sure they had access to inventory on a per impression basis elsewhere. To be strategically relevant, our competition needed to develop a solution to remain competitive with our full stack. This need was met in the form of header bidding. Developed initially by Criteo and adopted by other networks and exchanges, header bidding allowed these demand sources to submit prices into DFP and other ad servers on a (nearly) per impression basis. This increased competition across all direct and indirect demand sources and led to significant uplift for pubs.
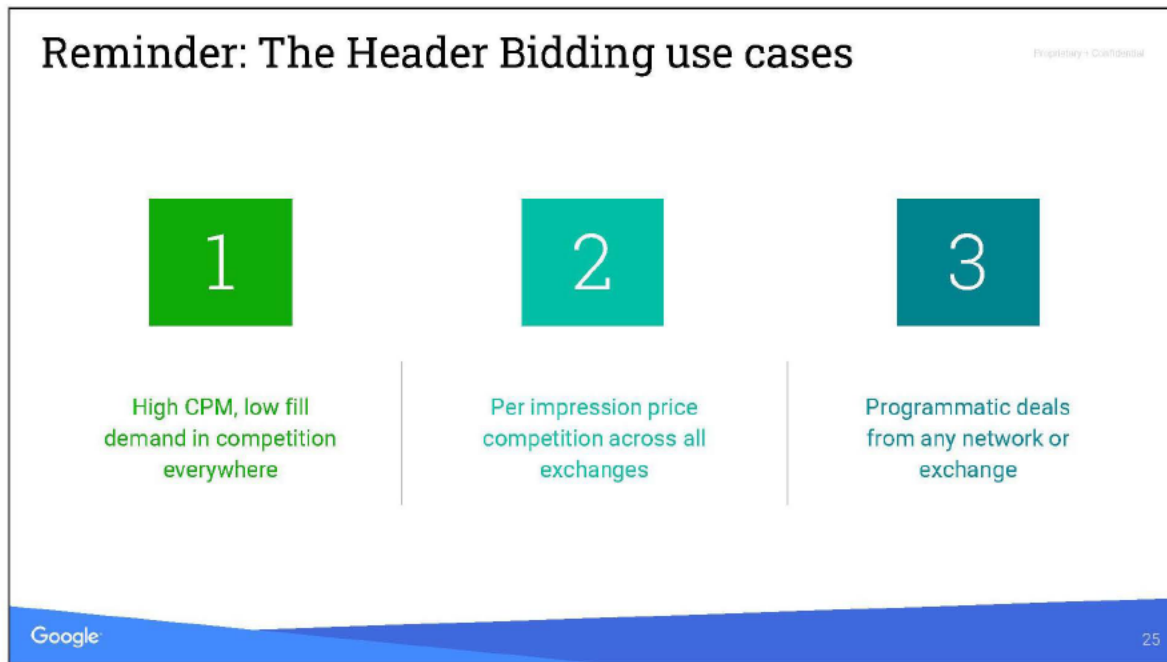
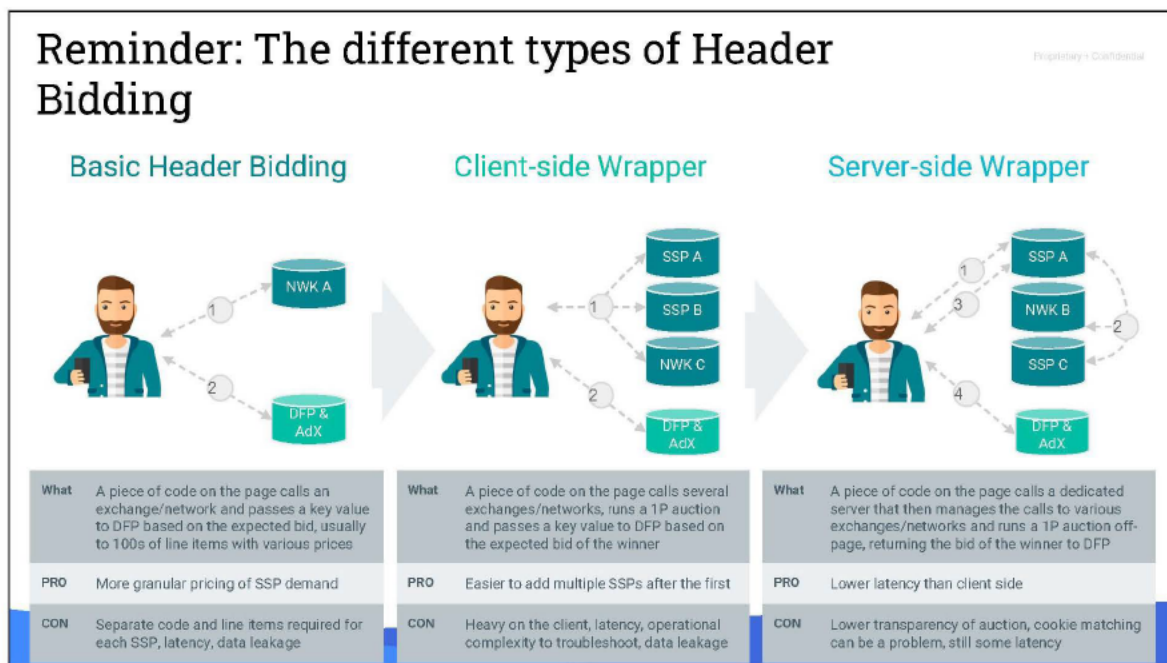Here are the three distinct use cases have developed due to the investment in header bidding.
High CPM, low fill demand (e.g. remarketing by Criteo and Amazon) is able to compete against the vast majority of standard direct and indirect impressions
Exchanges can compete against one another with accurate per impression pricing
Programmatic deals from any network or exchange can compete against traditional directly sold ads

Reminder: The different types of Header Bidding

Proprietary + Confidential

	Basic Header Bidding	Client-side Wrapper	Server-side Wrapper

	Basic Header Bidding		Client-side Wrapper		Server-side Wrapper
What	A piece of code on the page calls an exchange/network and passes a key value to DFP based on the expected bid, usually to 100s of line items with various prices	**What**	A piece of code on the page calls several exchanges/networks, runs a 1P auction and passes a key value to DFP based on the expected bid of the winner	**What**	A piece of code on the page calls a dedicated server that then manages the calls to various exchanges/networks and runs a 1P auction off-page, returning the bid of the winner to DFP
PRO	More granular pricing of SSP demand	**PRO**	Easier to add multiple SSPs after the first	**PRO**	Lower latency than client side
CON	Separate code and line items required for each SSP, latency, data leakage	**CON**	Heavy on the client, latency, operational complexity to troubleshoot, data leakage	**CON**	Lower transparency of auction, cookie matching can be a problem, still some latency

There are also different types of Header Bidding

Basic Header Bidding
Simple to work with one partner but separate codes if working with several SSPa, adding latency

Client-side Wrapper
Offers inventory to multiple exchanges before making ad calls to their ad servers
Ad calls are made from the user's browser and auction rules are also run by the browser using code in the header
Prone to higher levels of latency and impacted user experience
Examples: Rubicon Fast Lane, AppNexus

Server-Side Wrapper
Ad calls are made outside of the user's browser with the auction taking place in an external server after a single ad call is made
However, the browser still makes the first call to the header bidder server and receives an answer which it passes to the adserver
Improves speed and latency, but will not without any drawbacks (e.g. potential cookie loss)
Example: Amazon, Prebid.js Server

These are distinct from true server-to-server integration like Google's Exchange Bidding, which is NOT a form of header bidding (there is no code in the header)

Server to Server
All ad calls and auction rules are run off-browser with the tech provider managing and hosting the entire solution
There is no header code on the page and no initial call from the user's browser to the header bidding server. Communications flow directly between the publisher ad server and the other server.
Allows for a true unified auction to take place, with improved speed and reduced latency and inefficiencies for partners
Example: DoubleClick Exchange Bidding in Dynamic Allocation

While the technology in Header Bidding has become more sophisticated over time, it still requires at least one extra tag on the page. The additional client-side request(s) along with other server-side technology add latency for both publishers and users. Other problems exist throughout the end-to-end process (e.g. creative controls, pricing & billing transparency, data leakage) but publishers have been willing to put up with such issues in return for greater revenue.

On the buy-side, header bidding poses a different set of issues. When multiple exchanges are called for a single ad slot, buyers wind up bidding multiple times for the same impression, resulting in self-competition. The increase in queries-per-second (QPS) for both networks and DSPs add to machine costs. These problems gets worse when header bidding providers sub-syndicated impressions to other unauthorized SSPs that take a cut and reduce the share of an advertiser's dollar that makes it to the publisher. Beyond the pure machine costs and additional revenue shares or tech fees, engineering resources are needed to develop intelligent bid filtering and decision logic.